

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**МЕТОДИЧНІ ВКАЗІВКИ**  
до лабораторних занять та самостійної роботи  
**«МЕТОДИ КОНСТРУЮВАННЯ ОБ'ЄКТІВ  
В КОМП'ЮТЕРНИХ СИСТЕМАХ»**  
(Частина II)

для студентів спеціальності 122  
«Комп'ютерні науки»

Затверджено  
редакційно-видавничою  
радою університету,  
протокол № 2 від 24.05.2018 р.

Харків 2019

Методичні вказівки для студентів спеціальності 122 «Комп'ютерні науки».  
Методи конструювання об'єктів в комп'ютерних системах./ укл.  
І. Ю. Адашевська. – Х.: НТУ «ХПІ», 2019. – 36 с.

Рецензент: Л.М.Савченко.

Кафедра геометричного моделювання та комп'ютерної графіки

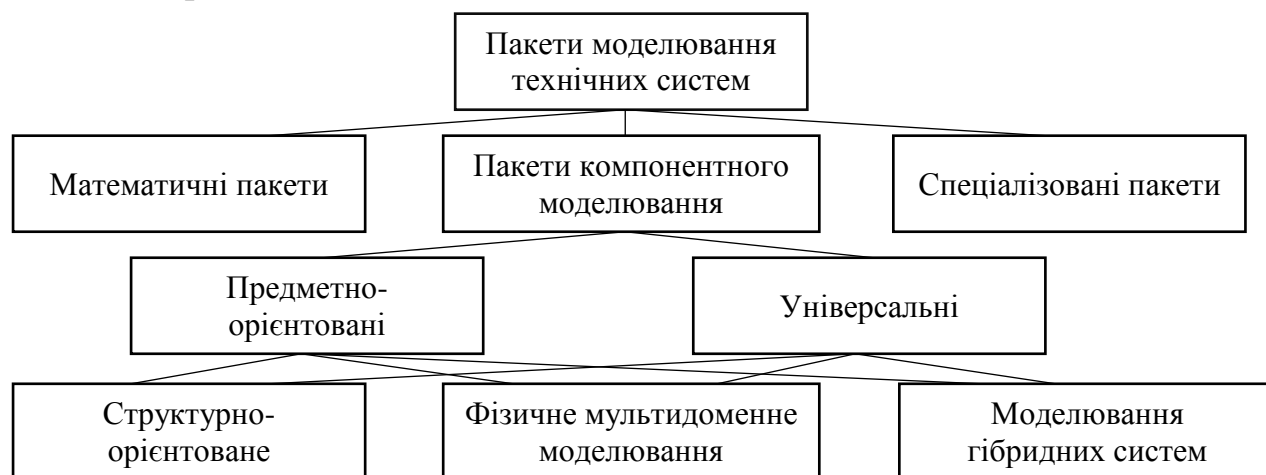
## Частина II. СИСТЕМИ ТЕХНІЧНОГО МОДЕЛЮВАННЯ

Поряд з розвитком цифрових обчислювальних машин формувалася напрямок аналогових обчислювальних машин (АОМ), за допомогою яких вирішувалися різні фізичні та математичні задачі. АОМ дозволяли вирішувати різні види математичних моделей, представлених у вигляді диференціальних рівнянь за допомогою натурного схемотехнічного моделювання. Аналогові ЕОМ в даний час не розробляють. Однак з'явилися технічні інформаційні системи (комп'ютерні віртуальні конструктори), зокрема Electronics Workbench, Vissim, LabVIEW, додаток Simulink системи MATLAB та інші системи, які вирішують математичні завдання за допомогою схемотехнічного моделювання.

Системи технічного моделювання побудовані за принципом конструктора, за системою блоків. У системах технічного моделювання можна вирішувати як математичні, так і інженерні завдання. У цих комп'ютерних системах можна збирати і конструювати віртуально будь-які електротехнічні схеми з використанням комп'ютерних аналогів електротехнічних і вимірювальних деталей, а також візуальне моделювання та конструювання інженерних, технічних імітаторів електронних приладів і логічних пристроїв. Більш того, спроектовані і створені віртуальні інженерні та виробничі комп'ютерні об'єкти та установки можна використовувати для натурного експерименту і виробничих випробувань в реальному масштабі часу.

Моделювання технічних систем – спрощене відображення реального виробу і його опису з метою оцінки відповідності його яким-небудь вимогам або здійснення вибору найкращого виробу з декількох альтернативних варіантів.

Класифікація пакетів моделювання технічних систем:



До числа універсальних, чи не орієнтованих на конкретні прикладні області пакетів для моделювання технічних систем можна віднести пакет MATLAB / Simulink, а також побудовані за його зразком та подібністю пакети VisSim, MBTU.

Ці пакети призначені для моделювання і дослідження динамічних систем в широкому розумінні цього терміну, включаючи і дискретні, і безперервні, і гібридні моделі. Їх відрізняє відносна простота та інтуїтивна ясність вхідних мов у поєднанні з розумними вимогами до потужності комп'ютерів.

## Лабораторна робота №1

**Тема:** конструювання об'єктів в технічній інформаційній системі.

**Мета роботи:** створення додатку, освоєння технології графічного програмування, практичний досвід у зміні та редагуванні властивостей графічних елементів управління та індикації в технічній інформаційній системі LabVIEW

### Теоретичні відомості.

LabVIEW (англ. Laboratory Virtual Instrumentation Engineering Workbench) – це середовище розробки і платформа для виконання програм, створених на графічній мові програмування «G» фірми National Instruments (США). LabVIEW використовують в системах збору та обробки даних, а також для управління технічними об'єктами і технологічними процесами.

Графічна мова програмування «G», яку використовують в LabVIEW, заснована на архітектурі потоків даних. Послідовність виконання операторів в таких мовах визначають не порядком їх слідування (як в імперативних мовах програмування), а наявністю даних на входах цих операторів. Оператори, не пов'язані даними, виконують паралельно в довільному порядку.

Програму LabVIEW називають і вона є насправді віртуальним приладом (англ. Virtual Instrument) та має дві частини:

- блокової діаграми, яка описує логіку роботи віртуального приладу;
- лицьової панелі, яка описує зовнішній інтерфейс віртуального приладу.

Віртуальні прилади можна використовувати в якості складових частин для побудови інших віртуальних приладів.

Лицьова панель віртуального приладу містить засоби введення-виведення: кнопки, перемикачі, світлодіоди, верньєри, шкали, інформаційні табло і т. п. Їх використовує людина для управління віртуальним приладом, а також іншими віртуальними приладами для обміну даними.

Блокова діаграма містить функціональні вузли, які є джерелами, приймачами і засобами обробки даних. Також компонентами блокової діаграми є термінали ( «задні контакти» об'єктів лицьової панелі) і керуючі структури (є аналогами таких елементів текстових мов програмування, як умовний оператор «IF», оператори циклу «FOR» і «WHILE» і т. П.).

Функціональні вузли та термінали об'єднані в єдину схему лініями зв'язків.

## Порядок виконання роботи

Запустіть програму National Instruments LabVIEW в наслідок чого відкриється вікно, як на рис. 1.1.

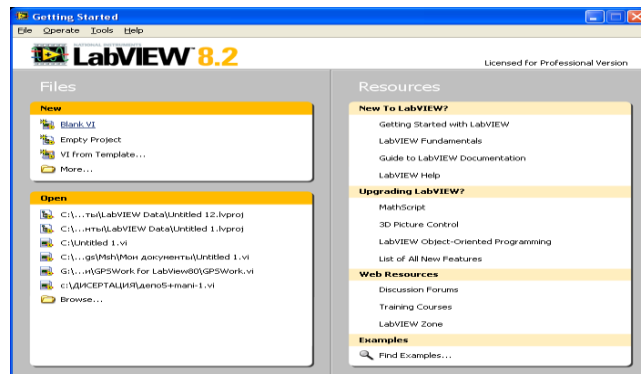


Рисунок 1.1 – Стартове вікно LabView

У вікні, що з'явилося виберіть опцію Blank VI. Якщо панель керування неактивна, то її слід викликати через основне меню Window -> Show Block Diagram. Для цього виконайте наступні кроки:

1. Використайте курсор миші у вигляді стрілки, встановіть його на інтерфейсну панель інструменту Numeric Control, який знаходиться на панелі управління (Controls) -> Modern -> Numeric.
2. Підпишіть його як «Число А». Для цієї мети підведіть покажчик до мітки, клікніть і просто наберіть необхідний текст.
3. Встановіть, за аналогією з попередніми двома кроками, і підпишіть, як «Число В» ще один Numeric Control. Це будуть поля введення параметрів.
4. Для відображення результату необхідно розмістити на інтерфейсну панель **Numeric Indicator**, який також знаходиться на панелі управління (Controls) -> Modern -> Numeric. Підпишіть його, як «Результат». Має бути приблизно так, як показано на рис. 1.2.

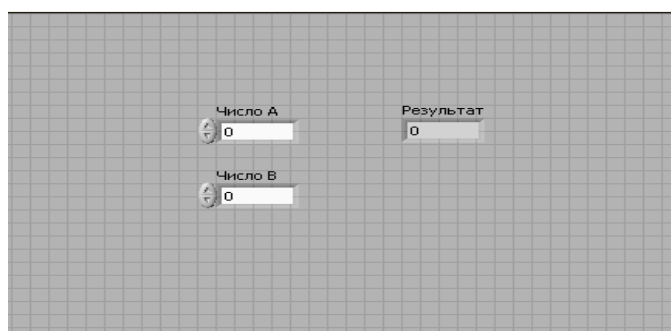


Рисунок 1.2 – Лицьова панель додатку

5. Тепер перейдемо до основної частини роботи, а саме до графічного програмування. На відміну від інших мов програмування, таких, як, наприклад, Borland Delphi або Microsoft Visual C ++, у цьому випадку не потрібно писати жодного рядка текстового коду, що реалізує певний алгоритм.

Створивши візуальний інтерфейс з двома полями введення чисел і одним цифровим індикатором, поставимо і реалізуємо завдання, наприклад, множення цих чисел. Для цього необхідно перейти в так зване вікно побудови діаграм, де можна побачити три іконки (терміналу), які відповідають полям введення чисел і індикатору. Реалізація простого або складного алгоритму буде зводитися до елементарної послідовності дій, тобто до установки необхідних іконок, які виконують ту чи іншу функцію і служать для зв'язку (з'єднання) їх між собою.

При множенні чисел необхідно викликати функціональну панель (Functions) і перетягнути трикутну іконку, що відповідає операції множення, до вікна редагування діаграм. Вона знаходиться в **Functions -> Programming -> Numeric -> Multiply**.

6. Тепер залишається з'єднати необхідні контакти сполучною катушкою (**Wiring Tool**). Підводимо покажчик миші до піктограми першого числа до тих пір, поки він не перетвориться у катушку, потім натискаємо ліву кнопку миші і, не відпускаючи її, з'єднуємо другий кінець лінії з одним з контактів піктограми перемноження. Для зміни напрямку зв'язку буде потрібно ще одне проміжне клацання лівою клавішею миші.

7. Повторюємо ці дії і для другого числа, аналогічно поєднуючи вихід іконки підсумовування з входом цифрового індикатора. Повинна вийти функціональна діаграма ( «текст» програми) (рис. 1.3)

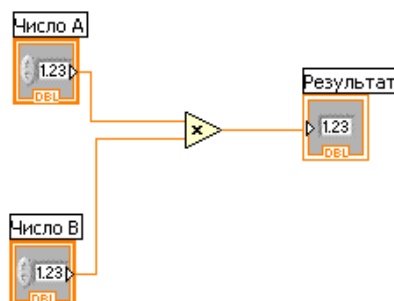


Рисунок 1.3 – Функціональна схема діаграми

Отже, програму написано. Тепер залишається запустити її на виконання і переконатися в її працездатності.

8. Переходимо на інтерфейсну панель, запускаємо програму на виконання в циклічному режимі, натиснувши лівою клавішею миші на кнопку циклічного запуску (**Run Continuously**).

Змінюємо значення полів введення чисел, використовуючи клавіатуру або мишу.

Для зупинки виконання програми слід скористатися кнопкою лінійки управління (**Abort Execution**).

Зробимо ще один крок і збережемо наш перший елементарний додаток на диску. Збереження **LabVIEW**-програми відбувається аналогічно запису, наприклад документа в Microsoft Word або Excel.

Для першого збереження програми необхідно вибрати в меню **File** пункт **Save**. У діалоговому вікні, що з'явилося необхідно вибрати або створити бажаний каталог (папку), ввести ім'я файлу і підтвердити введення. Записаний нами файл зберігся з розширенням **.vi** (**Virtual Instrument** – віртуальний інструмент) і має вигляд <ім'я файлу> **.vi**.

Файли з розширенням **vi** можна переносити між різними платформами, будь то Windows або Unix / Linux.

Змінимо зовнішній вигляд наших графічних об'єктів. Для цього підводимо курсор у вигляді стрілки на об'єкт, що відповідає «Числу А», і натискаємо праву кнопку миші.

У контекстному меню вибираємо опцію заміни (**Replace**). Далі входимо в підменю **Numeric** і там вибираємо кулясту ручку управління (**Knob**).

Змінимо розмір ручки. Змінюючи положення покажчика, побачимо, що в чотирьох точках він змінює вигляд зі стрілки на окружності. У цей момент, натиснувши і утримуючи ліву кнопку миші, змінюємо вигляд ручки.

Тепер розмістимо мітку «Число А», виділяючи і переміщуючи її в необхідну позицію.

Змінимо атрибути ручки «Число А». У меню, що з'являється, натискаючи праву кнопку миші, виберемо зміну візуальних властивостей об'єкта (**Visible Items**), а в них **Ramp**.



Для точного позиціонування ручки або відображення значення виберемо ще й властивість **Numeric Display**, яке також знаходиться в **Visible Items**. Додайте його на панель на ваш розсуд.

Проробіть аналогічні кроки і для інших елементів інтерфейсу. В результаті повинні отримати такий зовнішній вигляд, як на рис. 1.4.

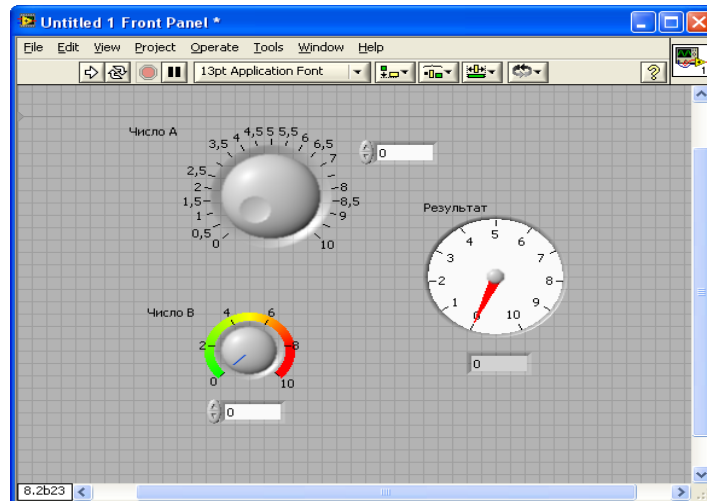


Рисунок 1.4 – Лицьова панель додатку

*Примітка.* Щоб змінити діапазон значення, яке ми вводимо або виводимо, необхідно в інструментах (**Tools**) вибрати можливість редагування тексту (**Edit Text**), підвести курсор до початкового або кінцевого діапазону шкали, натиснути ліву кнопку миші і змінити значення на необхідне.

Трохи ускладнимо завдання. Встановимо на панель графічний екран, який буде відображати графічну залежність значення суми чисел від часу. Для цього на панелі управління (**Controls**) виберемо іконку **Graph**, а у підменю, що з'явилося **Waveform Chart**.

Встановлюємо графічний екран на панель і змінюємо його атрибути і властивості так, як показано на рис. 1.5.

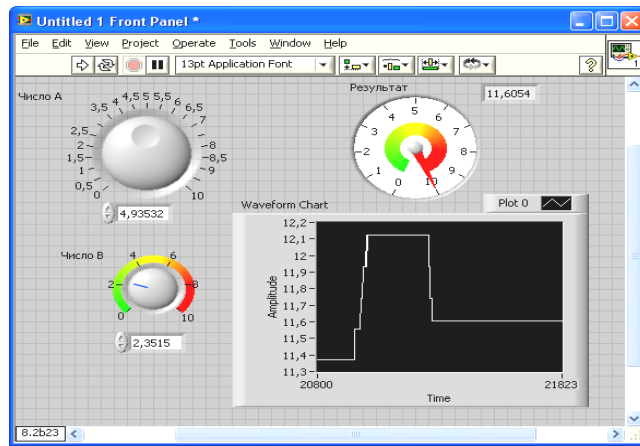


Рисунок 1.5 – Лицьова панель додатку з графіком тимчасової залежності

Тепер продовжимо розробку нашого програмного коду. Встановимо зв'язок між значенням суми і **Waveform Chart**. Для цього перейдемо у вікно діаграм і з'єднаємо за допомогою катушки (**Wiring Tool**) необхідний контакт з відповідною іконкою графічного вікна (рис 1.6).

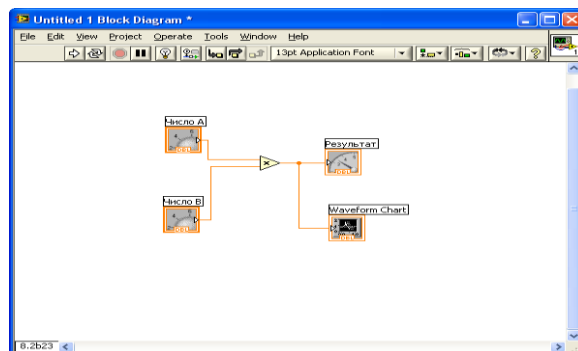


Рисунок 1.6 – Функціональна схема діаграми з тимчасовою залежністю

До цього часу ми запускали програму на виконання в циклічному режимі. Тепер «розмістимо» фрагмент нашої діаграми в цикл, наприклад, такий, який є аналогом оператора «**While**». Умовою виходу з циклу буде натискання на кнопку зупинки. Для реалізації поставленого завдання необхідно виконати наступні кроки.

У функціональному меню (**Functions**) виберіть структури (**Structures**), а в них цикл **While-Loop**. Перетягніть його у вікно редагування діаграм, збільшіть цикл **While-Loop** до розміру, який повинен бути трохи більше ніж створена вами раніше структурна схема (рис. 1.7).

Розмістіть базову структуру в цикл **While-Loop**. Для цього покажчиком у вигляді стрілки виділяємо всю схему (утримуючи ліву кнопку миші, виділяємо

прямокутну область, куди потрапляють під виділення всі необхідні елементи схеми), а потім перетягуємо виділені елементи всередину циклу **While-Loop**.

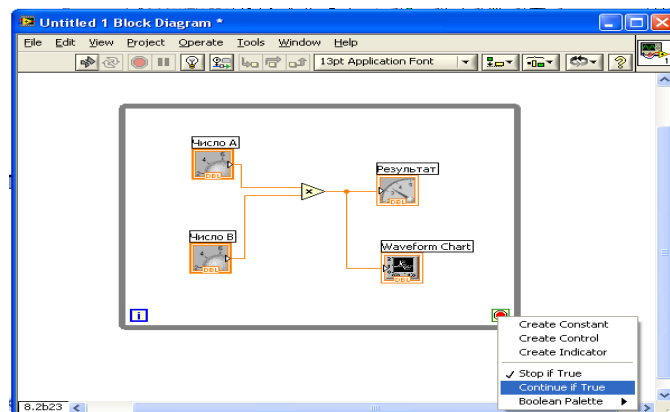


Рисунок 1.7 – Функціональна схема діаграми з використанням циклу **While-Loop**

Змінимо вигляд іконки умови виходу з циклу **While-Loop**, натискаючи ліву кнопку миші на відповідній піктограмі таким чином, як це показано на рис. 1.7.

Тепер натискаємо на тій самій іконці, але вже правою кнопкою миші. У меню, що з'являється, слід вибрати пункт створення керуючого елемента (**Continue if True**). В результаті з'явиться піктограма, що відповідає кнопці «Зупинка».

Створення будь-якого елемента управління або індикатора у вікні редагування діаграм тягне за собою створення відповідного графічного компонента на головній інтерфейсній панелі.

Для того, щоб зміна значень суми, яку виводять у вигляді графіка, виконувалась з певною затримкою в часі (для більшої наочності аналізу роботи), встановимо з функціональної панелі іконку

**Functions> Time & Dialog> Wait Until Next ms Multiple.**

Створимо для неї ручку контролю, відповідну цій затримці. Для цього підведемо покажчик у вигляді катушки до лівої частини іконки, натиснемо праву кнопку миші і виберемо пункт створення константи

**(Create> Control)** (рис. 1.8).

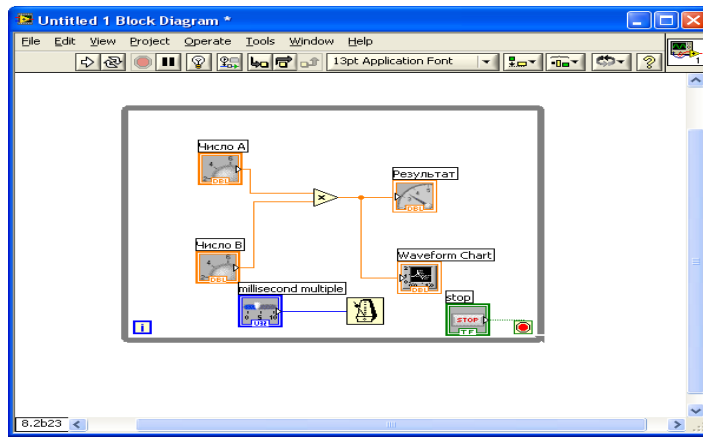


Рисунок 1.8 – Функціональна схема діаграми з затримкою

Залишається тільки сформувати передню інтерфейсну панель. Як і для будь-яких візуальних приладів, регуляторів, повзунків, меню і т. д., для **Waveform Chart** також можна змінювати різні візуальні параметри і властивості. Запускаємо програму на виконання. В результаті повинні отримати результат, представлений на рис. 1.9.

Натиснувши на створену нами кнопку «СТОП», ми зупинимо програму.

Переходимо на панель відображення і робимо перекомпонування елементів на панелі, змінюємо товщину і колір лінії промальовування графіка.

Редагування властивостей графічного екрану **Waveform Chart** (зокрема, колір і товщина лінії) здійснюється через узагальнене меню його властивостей, виклик якого відбувається простим натисканням миші інструментом **Operation Tool** в області верхнього напису (рис. 1.10).

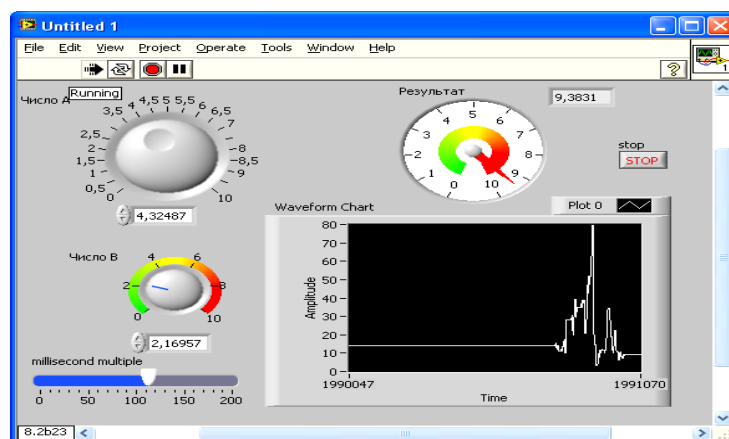


Рисунок 1.9 – Лицьова панель діаграми з використанням циклу

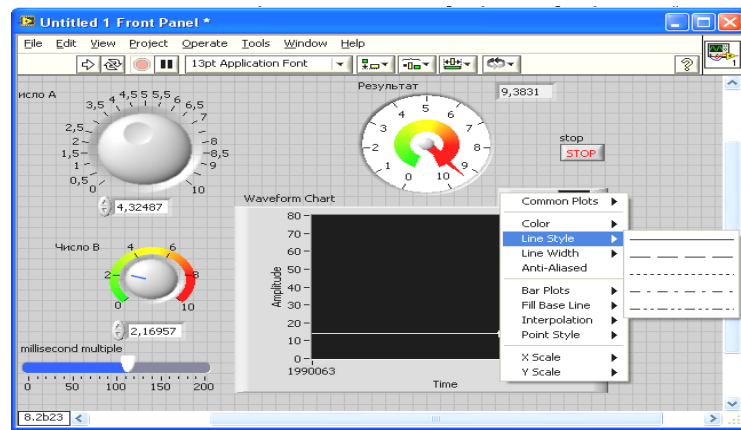


Рисунок 1.10 – Зміна властивостей графіка

## Лабораторна робота №2

**Тема:** конструювання об'єктів в технічній інформаційній системі

**Мета роботи:** вивчення середовища моделювання **VisSim (Visual Simulator)** та конструювання найпростіших моделей.

**Теоретичні відомості.**

**VisSim** дозволяє робити побудову, дослідження та оптимізацію віртуальних моделей фізичних та технічних об'єктів, в тому числі і систем управління.

**Умова:**

У повітря з однієї точки запущені два об'єкти. Траєкторію одного описують параболою виду  $y_1 = 5G \cdot x^2$  ( $y_1$  – висота, м;  $x$  – час, с). Траєкторія іншого – це синусоїда з частотою 4 Rad/s і амплітудою 5 м, що проходить уздовж прямої  $y = 8x$  ( $y$  – висота направляючої, м). Скільки разів траєкторії об'єктів перетнуться, і на якій висоті за 30 с спостережень?

**Порядок виконання роботи**

1. Встановлюємо і запускаємо середовище візуального моделювання VisSim.

2. Додамо до схеми наступні блоки: генератор «синусоїда» зі значенням амплітуди 5 м. і частотою 4 рад / с відповідно до завдання, генератор сигналу, що змінюється з постійною швидкістю з посиленням сигналу до 8 разів і генератор «парабола» з коефіцієнтом  $0.11 \cdot 5 = 0.55$  (відповідно до варіанту). До того ж виходи останніх двох блоків подамо на вхід суматора для реалізації траєкторії руху другого об'єкта, що представляє собою синусоїду, що проходить уздовж прямої  $y = 8x$ . Задамо наступні параметри симуляції: часовий інтервал від 0 до 30 секунд, крок симуляції 0.01. Результати представлені на рис. 2.1.

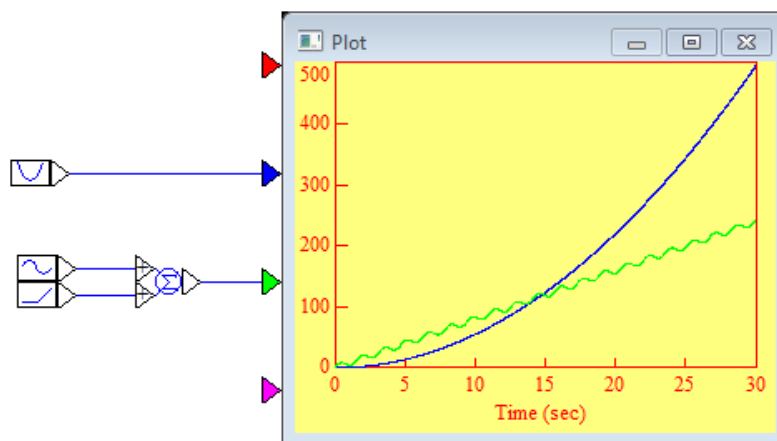


Рисунок 2.1 – Схема, що моделює траєкторії руху двох тіл

3. На представленій осцилограмі можна побачити, що траєкторії руху тіл перетнуться приблизно за  $t = 15$  с. Для однозначного і точного визначення часу і висоти перетину траєкторій задаємо часовий інтервал від 13.5 до 15 с. Результат представлено на рис. 2.2.

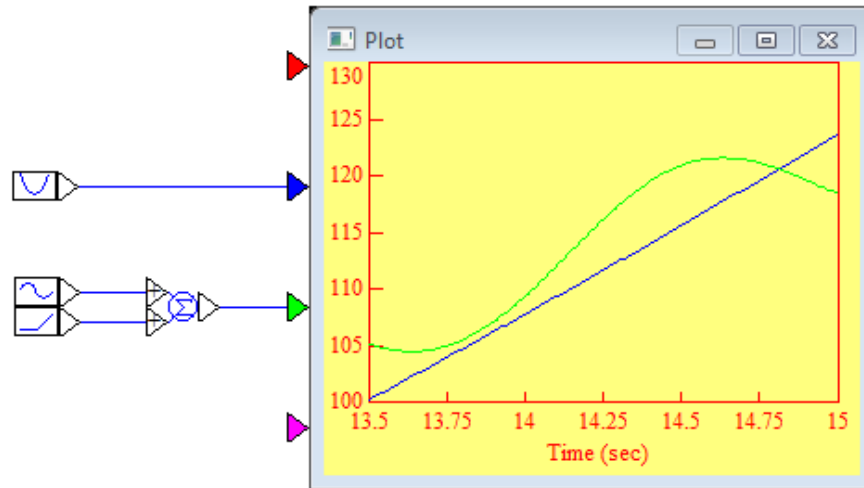


Рисунок 2.2 – Траєкторії руху двох тіл у період з 13.5 до 15с.

З рисунку можна побачити, що траєкторії руху тіл перетнуться лише один раз. Для якомога точнішого визначення часу і висоти перетину, задаємо часовий інтервал 14.8-14.82 с. Осцилограма представлена на рисунку 2.3

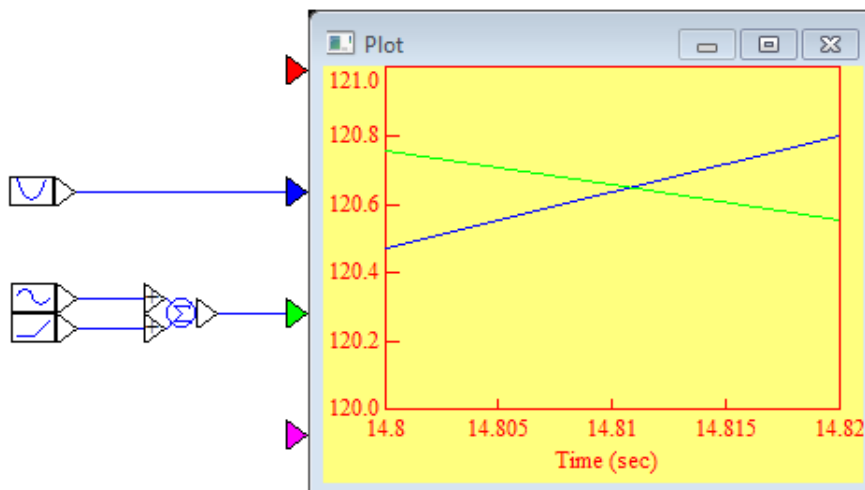


Рисунок 2.3 – Траєкторії руху тіл на інтервалі 14.8–14.82 с

Тепер можна зробити остаточний висновок, що траєкторії руху двох тіл перетнуться лише один раз за  $t \approx 14.812$  с. на висоті  $\approx 120.65$  м.

VisSim дозволяє робити побудову, дослідження та оптимізацію віртуальних моделей фізичних та технічних об'єктів, в тому числі і систем управління. Блоки VisSim'а генератор Парабола (parabola) та осцилограф (plot) дійсно є віртуальними аналогами реальних пристроїв.

## Лабораторна робота №3

**Тема:** конструювання об'єктів в технічній інформаційній системі.

**Мета:** навчитися працювати інструментами інтерфейсу і проектувати примітивні електронні пристрої в програмному пакті **P-CAD**, створювати друковані твердотілі плати.

### Теоретичні відомості

Система P-CAD 2000 призначена для наскрізного проектування електронних пристроїв і є лідером на ринку таких інструментальних засобів проектування.



Запуск програм можливо здійснювати зі стартового меню Windows.

За допомогою цього графічного редактора P-CAD Schematic створюють принципову схему проектованого пристрою. Крім того, з нього можна запускати й інші програмні модулі пакета.

За наявності файлів, створених цим редактором, досить виділити один з них і двічі клацнути мишею на його імені. Найбільш правильно це буде робити з самого графічного редактора. Зайдіть в меню File / Open ..., після чого в списку схемних файлів (вони мають розширення. Scft) виберіть потрібний.

Бажаючи запустити графічний редактор стандартними засобами Windows, необхідно двічі клацнути на обраному SCH-файлі. Відкриється діалогова панель «Додаток», натискаємо на ній кнопку “Інша” ..., відшукуємо програму схемного редактора, (вона має назву Scft. Exe і знаходиться в папці PCAD 2000 /) і запускаємо її. Після першого звернення, її буде додано до списку програм для відкриття файлів, і при повторному викликанні цієї програми не треба буде робити зайвих дій.

### Порядок виконання роботи

1.  Запускаємо графічний редактор одним із засобів, які описано вище. На екрані монітора з'явиться робоче вікно (рис.3.1), основну частину якого займає область малювання. У верхній частині розташовані вікна меню команд, а трохи нижче – панелі інструментів, на яких розміщені команди, які ми найчастіше використовуємо, наприклад PlacePart (Розмістити компонент), або  PlaceWire (Провести провідник).



Розташована горизонтально панель містить піктограми системних команд, панель команд розміщення розташована вертикально). Найбільшу частину екрана займає робоче вікно, призначене для введення креслень схем.

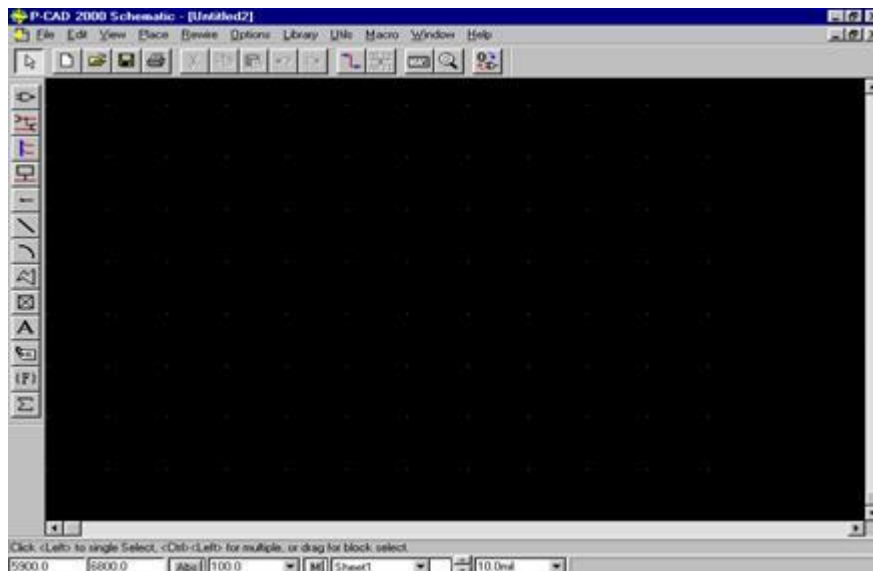


Рисунок 3.1 – Робоче вікно

бібліотеку, знайти в ній потрібний компонент і розмістити його на схемі. У нижній частині робочого вікна розташовано рядок повідомлень. У найнижчій частині екрана розташовано рядок станів.

2. Розглянемо найбільш використовувані команди панелі розміщення. Всі піктограми, розташовані на цій панелі, продубльовані в меню Place. При натисканні на піктограму (PlacePart) з'являється діалогове вікно (рис3.2), за допомогою якого можна підключити

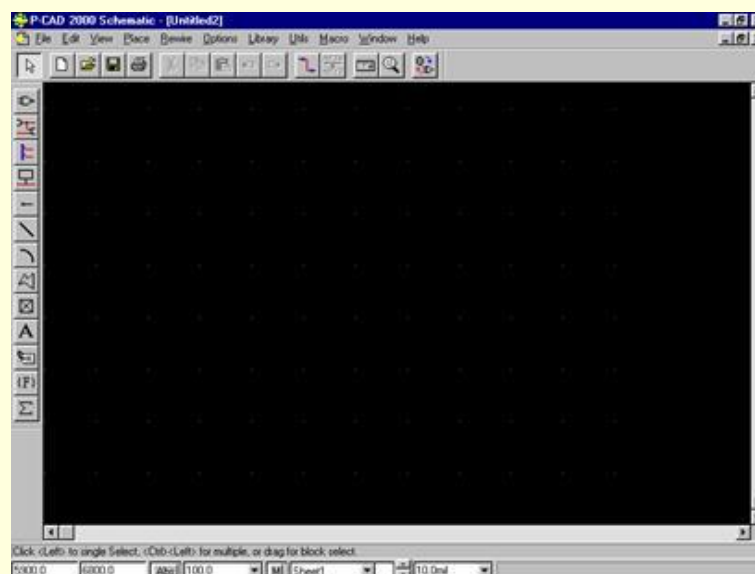


Рисунок 3.2 – Діалогове вікно PlacePart

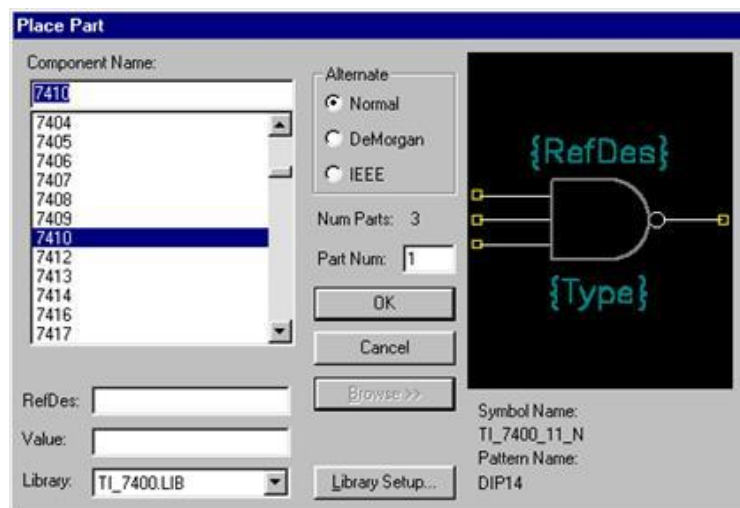



Рисунок 3.3 – Схема в редакторі Schematic

3.  Наступні піктограми служать для розміщення провідників (PlaceWire) і шин (PlaceBus). З іншими піктограмами познайомимося пізніше. Тепер спробуємо намалювати просту схему, і для прикладу візьмемо схему, побудовану на елементах серії K555, представлену на рис. 3.3.

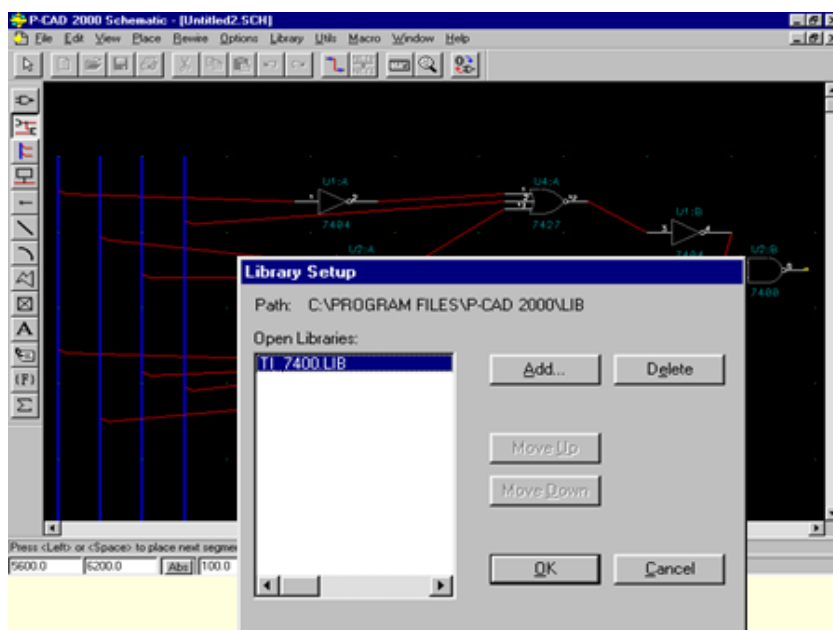


Рисунок 3.4 – Діалогове вікно LibrarySetup

Клацаємо на піктограмі. З'являється діалогове вікно (рис. 3.2), в якому знаходяться компоненти, доступні для розміщення. Перш за все, необхідно підключити до нашого проекту бібліотеки, що містять потрібні компоненти. Установка бібліотек відбувається натисканням на кнопку LibrarySetup ..., після чого з'являється діалогове вікно, представлене на рис. 3.4.

Ліворуч розташовано список підключених бібліотек, вгорі місце розташування бібліотеки. Натиснемо кнопку «Add» ..., зі списку виберемо бібліотеку

Ti\_7400.lib (швидше за все вона розташована в каталозі \ ProgramFiles \ P-CAD 2000 \ Library). Далі натискаємо «ОК» і повертаємося в попереднє діалогове вікно (рис. 3.2).

Список доступних компонентів можна переглянути за допомогою смуги прокрутки. Вона містить активовану бібліотеку, у нашому випадку це Ti\_7400.LIB.

4. Тепер вибираємо зі списку компонентів потрібні нам і знаходимо за таблицею мікросхем відповідні аналоги: ЛН1 (7404), ЛА3 (7400), ЛА4 (7410), ЛЕ4 (7427), ЛЕ1 (7402). У цьому ж вікні ми можемо отримати графічну інформацію про компоненти. Для цього натискаємо на кнопку Browse>, і перед Вами з'являється умовне графічне позначення обраного елемента. Зверніть увагу, що редактор видає інформацію про ім'я символу компонента *SymbolName* та ім'я його посадкового місця *PatternName*; якщо інформація про посадкове місце відсутня – це означає, що цей компонент не можна розміщувати на друкованій платі в редакторі РСВ.

Натискаємо «ОК». Для того, щоб зафіксувати положення елемента на робочому полі проекту треба клацнути лівою кнопкою миші. Звернімо увагу, що поруч з його графічним зображенням з'явилося ім'я U1. Переміщуємо курсор в інше місце, і знову натискаємо ліву кнопку. На екрані з'являється ще один елемент з ім'ям U2. Таким чином, можна розміщувати скільки завгодно копій, поки ви не натиснете праву кнопку миші або Esc.

5. Для видалення непотрібного нам елемента, необхідно увійти в режим вибору об'єктів, тобто клацнути піктограму або натиснути клавішу S на клавіатурі. Тепер клацаємо лівою клавішею миші по елементу, за замовчуванням виділений елемент позначають жовтим кольором. Для переміщення елемента, не відпускаючи ліву кнопку миші, пересуваємо його в потрібне місце. Для того, щоб видалити елемент, натискаємо клавішу Del, і елемент зникає з екрану.

Тепер розмістимо елементи схеми, які залишилися. Знову клацаємо піктограму PlacePart і вибираємо зі списку наступний компонент. Так треба повторювати до того часу, поки схему не буде завершено.

Для того щоб повернути елемент потрібно його виділити клацанням лівої кнопки миші і натиснути клавішу «R».

6. Тепер ми отримали на робочому полі всі цеглинки для нашої схеми, але щоб отримати закінчену схему, все елементи потрібно з'єднати провідниками. Для цього треба клацнути піктограму PlaceWire. Для того, щоб намалювати ланцюг, який кілька разів змінює напрямок, необхідно в точках зламу фіксувати вже намальовану частину провідника клацанням лівої кнопки миші. Щоб закін-

чити малювання ланцюга, натискаємо праву кнопку миші або Esc. Тепер подивимося, чи не залишилося у нас не приєднаних ланцюгів. «Висячі» ланцюги закінчуються жовтим квадратом. Якщо ми все робили правильно, робоче поле проекту має відповідати рис.3.3

Тепер настав час зберегти наш проект. Входимо в меню Files \ Save ..., вказуємо папку, де буде розміщено наш файл, і присвоюємо йому ім'я, наприклад urok. sch.

Як ми зазначили вище, P-CAD 2000 – система наскрізного проектування. І тому робота з цією програмою не закінчується створенням принципової схеми.

7. Для того, щоб передати дані про виріб, який ми проектуємо, на друковану плату, необхідно з наявного у нас файлу принципової електричної схеми створити список з'єднань. Заходимо в меню Utils і запускаємо команду GenerateNetlist ... (рис. 3.5).

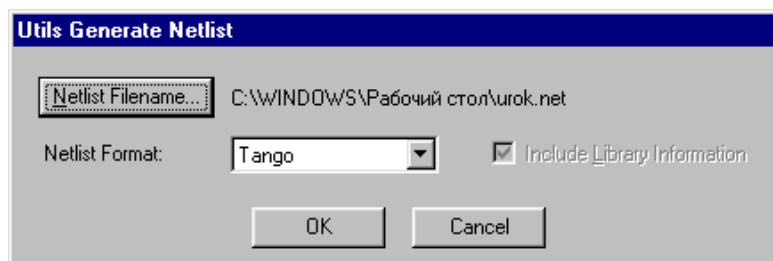


Рисунок 3.5 – Створення списку з'єднань

Натискаючи на кнопку NetlistFilename ..., ми зможемо вказати розташування і ім'я файлу списку з'єднань. У списку, що з'являється, вибираємо формат списку з'єднань, в нашому випадку це буде Tango. При натисканні на кнопку ОК запускається утиліта генерації списку.

Тепер необхідно створити архівну бібліотеку, в якій містяться всі компоненти, які ми використовуємо в нашому проекті. Вибираємо в меню Library команду ArchiveLibrary, зазначаємо розташування і ім'я бібліотеки: urok1. lib. Тепер ми маємо все для створення файлу друкованої плати.

8. Запускаємо редактор P-CAD 2000 PCB. Перш за все, ми повинні підключити бібліотеку, яка містить необхідну інформацію про компоненти проекту. Робимо це в такий спосіб: в меню «Бібліотека» вибираємо команду «Налаштування», з'являється вже знайоме нам діалогове вікно (рис.3.4). Додаємо створену нами бібліотеку проекту-urok. lib. Після підключення бібліотеки можемо приступити до завантаження файлу з'єднань. У меню «Утіліти» запускаємо команду: «Завантажити список з мережі», з'являється діалогове вікно (Рис.3.6).

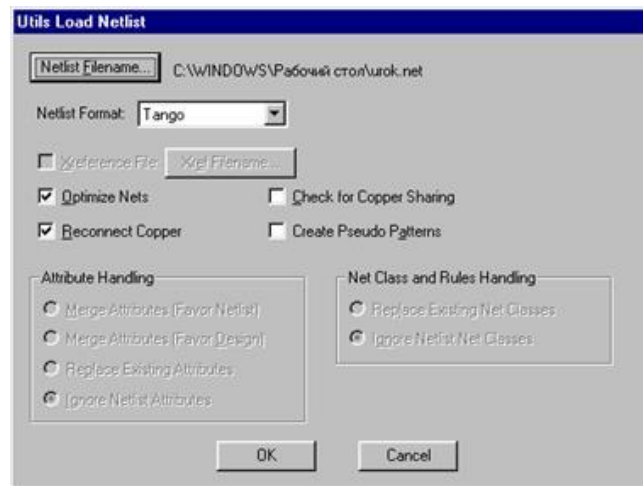


Рисунок 3.6 – Завантаження списку з'єднань

9. Натиснувши на кнопку NetlistFilename ..., вказуємо розташування та ім'я файлу списку з'єднань urok.net. У полі NetlistFormat вказуємо формат цього списку. В результаті на робочому полі РСВ повинні з'явитися компоненти проекту, з'єднані лініями зв'язку (НЕ розведеними провідниками, за термінологією P-CAD) (рис. 3.7).

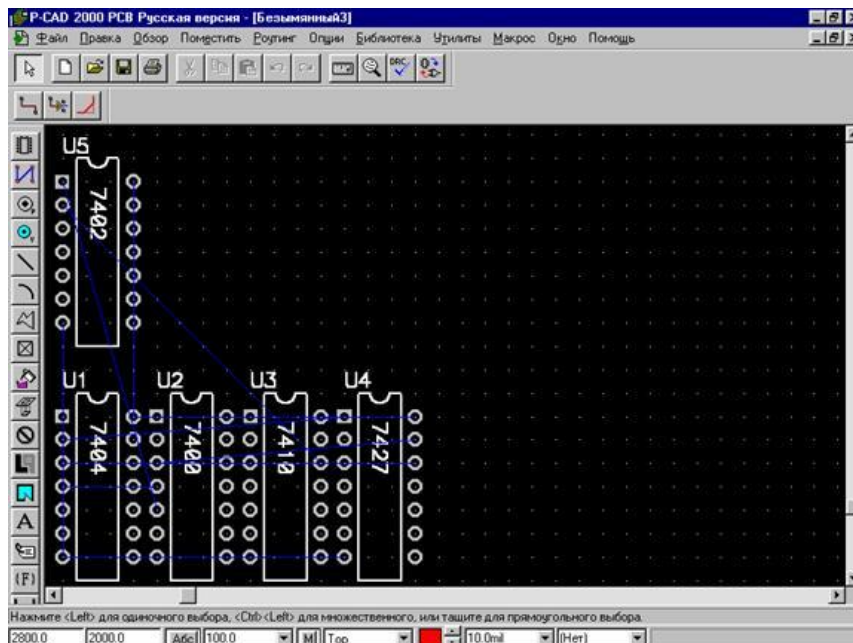


Рисунок 3.7 – Вікно графічного редактора РСВ

Для розведення друкованої плати, необхідно намалювати її контур в призначеному для цього шарі Board. Зробимо цей шар поточним. В меню Опції вибираємо команду Шари. З'являється діалогове вікно (рис.3.8), і в списку CurrentLayer вибираємо Board, потім Close.

10. Отже, тепер на робочому полі проекту необхідно нанести контур друкованої плати. За командою: “Розмістити лінію” або, клацнувши по піктограмі, \ намалюємо прямокутний контур. Щоб намалювати лінію, яка кілька разів міняє напрям, необхідно в точках зламу фіксувати вже намальований відрізок клацанням лівої кнопки миші. Для завершення процесу малювання лінії, натискаємо праву кнопку миші або Esc. Треба перевірити, чи всі компоненти розташовані всередині контуру. Переміщати або видаляти компоненти з друкованої плати можна тим самим способом, що і під час редагування схеми в Schematic.

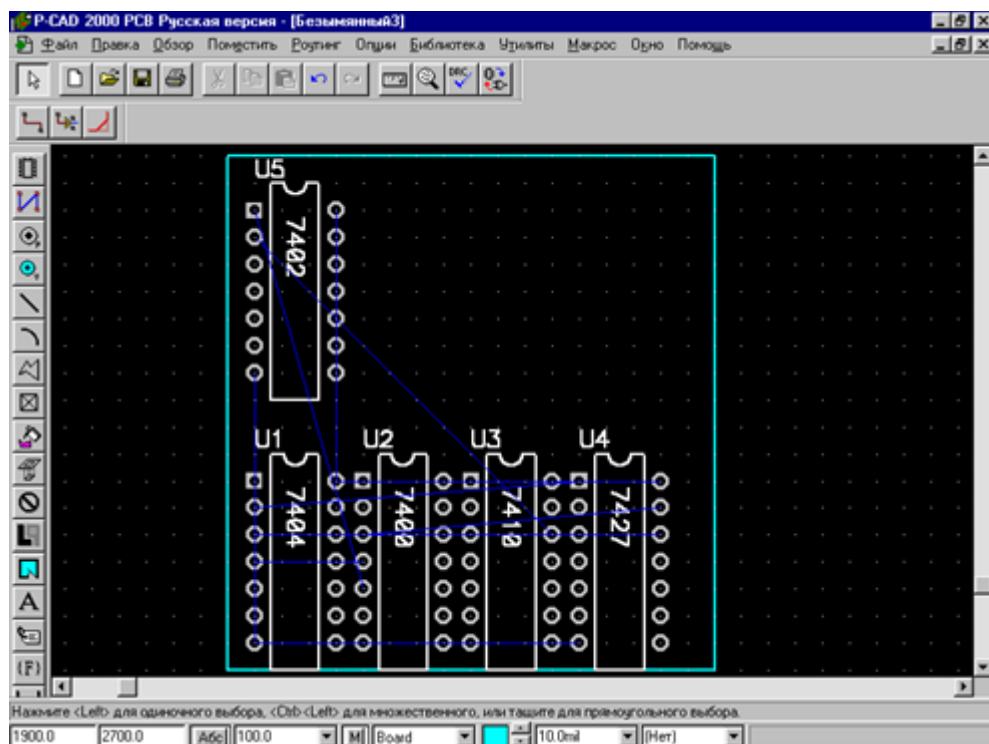


Рисунок 3.8 – Діалогове вікно OptionsLayers

Таким чином ми підготували друковану плату до автоматичного трасування. В меню «Роутінг» вибираємо команду «Автороутери» ..., в діалоговому вікні (рис.3.9) обираємо тип автотрасувальника QuickRoute. Це найпростіший з трасувальників, що входять до складу пакета P-CAD 2000. Його використовують лише для розведення нескладних плат, що містять невелику кількість компонентів, і для професійної розводки їх не рекомендують. Натискаємо кнопку Start, і на робочому полі редактора PCB з'являється вже розведена друкована плата. Проходить автоматична зміна ім'я файлу, на початку назви бачимо доданий префікс R. Тепер наш файл має назву Rurokl.pcb.

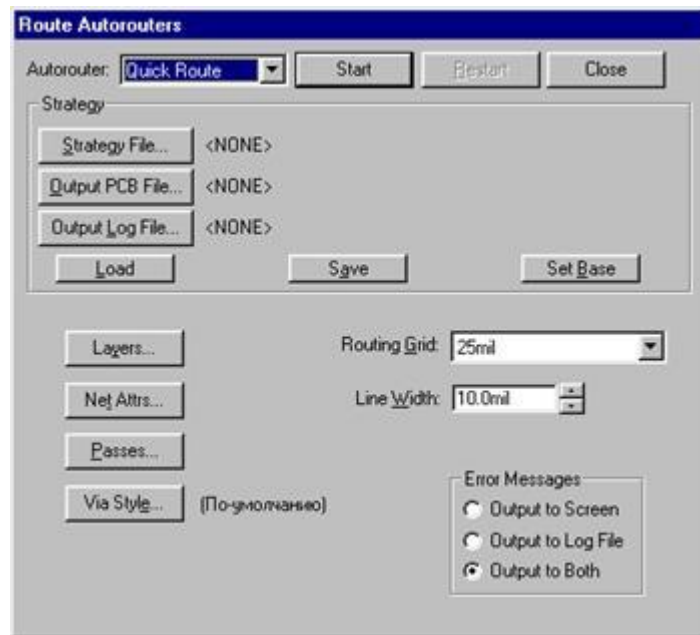


Рисунок 3.9 – Діалогове вікно Автороутери

На цьому процес створення нашої мікросхеми завершено.

11. Тепер ми можемо перевірити відповідність друкованої плати нашій принциповій схемі. З цією метою запускаємо одночасно два файли: файл принципової електричної схеми `ugok1.sch` і файл друкованої плати `Rugok1.pcb`. Якщо ми в режимі вибору об'єкта клацнемо по компоненту правою клавішею миші і в контекстному меню виберемо команду **Highlight**, то підсвіченими стануть і умовне графічне позначення на схемі, і посадочне місце на друкованій платі.

Таким чином, здійснюється гарячий зв'язок (рис.3.10) між редакторами Schematic і PCB.

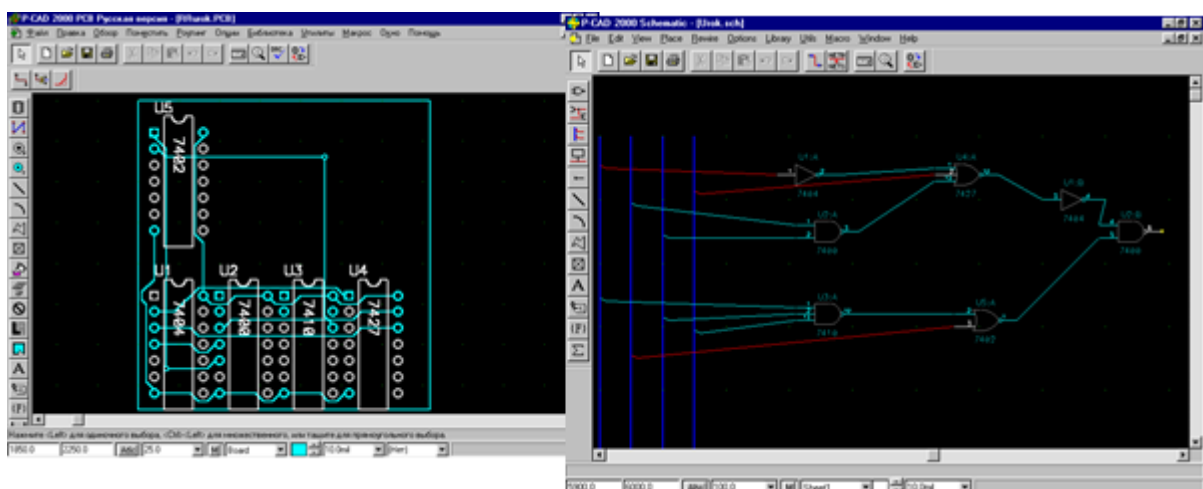


Рисунок 3.10 – Гарячий зв'язок

## Лабораторна робота №4

**Тема:** конструювання об'єкта з застосуванням технічної інформаційної системи моделювання.

**Мета роботи:** моделювання схеми засобами пакета Design Lab 8.0.

**Теоретичні відомості.**

Найбільшого поширення серед розробників радіоелектронної апаратури отримала система DesignLab корпорації MicroSim. До складу системи Design Lab входять наступні програми:

Структурна схема системи Design Lab представлена на рис 4.1

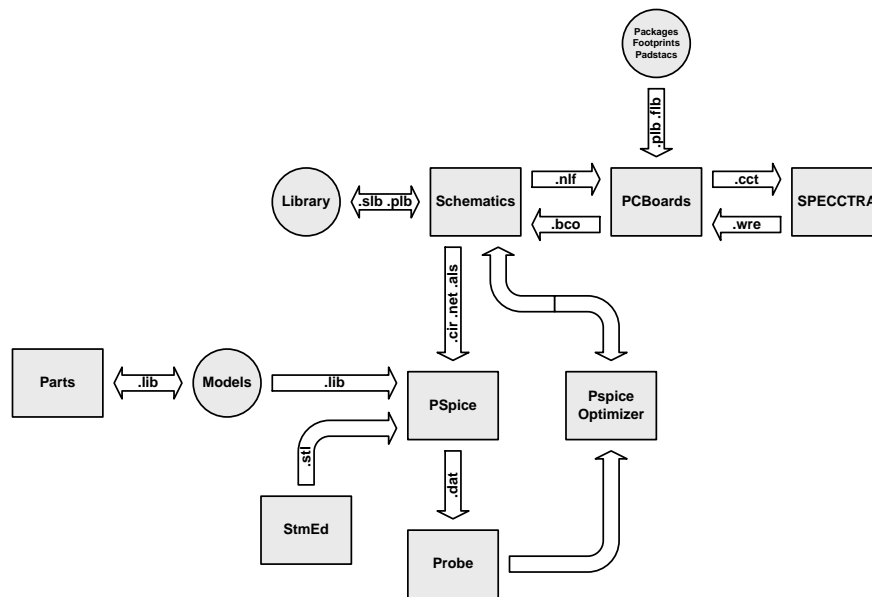


Рисунок 4.1 – Структурна схема системи Design Lab

Роботу з системою Design Lab зазвичай починають зі створення принципової схеми за допомогою редактора схем Schematics (рис.4.2). Він дозволяє створювати креслення принципів схем і здійснювати запуск інших програм (PSpice, Probe і ін.).



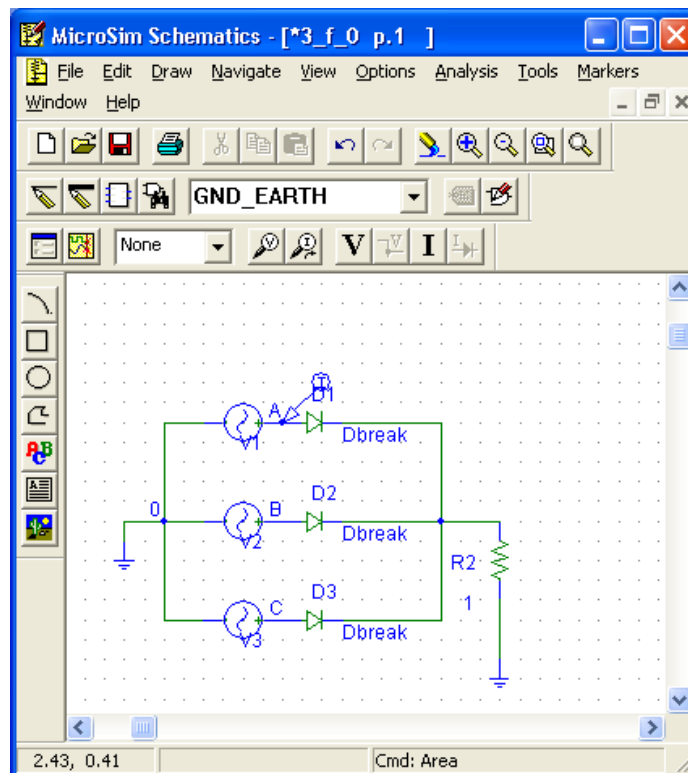



Рисунок 4.2 – Вікно програми Schematics


Графічний редактор викликають клацанням лівої кнопки миші по піктограмі Schematics в меню Пуск \ Програми \ Design Lab Release 8.0 \ Schematics. У процесі його завантаження відбувається підключення бібліотеки графічних символів компонентів. Редактор Schematics має основне меню і панель інструментів. Вбудована допомога дозволяє отримувати коротку інформацію про основні правила роботи (розділ Help в меню команд). На схему можна наносити символи кирилиці, застосовуючи шрифти True Type.

У верхній частині екрану розташовано горизонтальне меню, склад пунктів якого залежить від обраного режиму редагування (редагування принципів електричних схем або редагування символів), тобто умовних графічних позначень компонентів.

Спочатку вибираємо курсором команду File, після чого в меню, яке з'явиться, вибираємо рядок New, якщо створюємо нову схему, або рядок Open, якщо треба завантажити існуючу схему.

Символи компонентів наносимо на схему за командою Draw \ Get New Part (а також клацанням піктограми ). Обертання символу зручно проводити комбінацією клавіш [Ctrl] + [R], дзеркальне відображення – [Ctrl] + [F].

Останні обрані символи зручно наносити зі списку графічного меню. Для кожного компонента можна задати один або кілька параметрів (обрати його подвійним клацанням лівої кнопки миші), перелік яких зазначають заздалегідь при створенні його умовного графічного позначення. Конкретні значення параметрів призначають за допомогою діалогового вікна редагування атрибутів.

З'єднувальні дроти наносять «олівцем» – клацання піктограми *Draw wire* .

Для підготовки схеми до моделювання необхідно в середовищі редактора Schematics задати директиви моделювання за командою Analysis \ Setup, де встановлюють часовий інтервал і крок підрахунку параметрів моделювання.

Програму моделювання PSpice запускають командою Analysis \ Simulate або натисканням на клавішу *[F11]*. У верхній частині екрану розміщено рядок меню для завантаження файлів, зміни кольору та шрифту, виклику підказки. Необхідно відкрити файл з тим самим ім'ям, яким названо схему.

Потім у вікні програми PSpice з'являється назва завдання на моделювання, ім'я файлу, назва виконуваної команди і значення змінних параметрів, температури та ін. У середній частині екрану вказують назву виду аналізу, і наводять інформаційні повідомлення про фази виконання завдання на моделювання і повідомлення про помилки (останні виділяють червоним кольором). У нижній частині екрану в реальному масштабі часу виводять поточні значення від однієї до трьох змінних, перерахованих в директиві WATCH [на схемі цю директиву задають установкою символу у вигляді ока], й інформацію про крок зміни незалежних змінних і діапазон їх значень (рис.4.3).

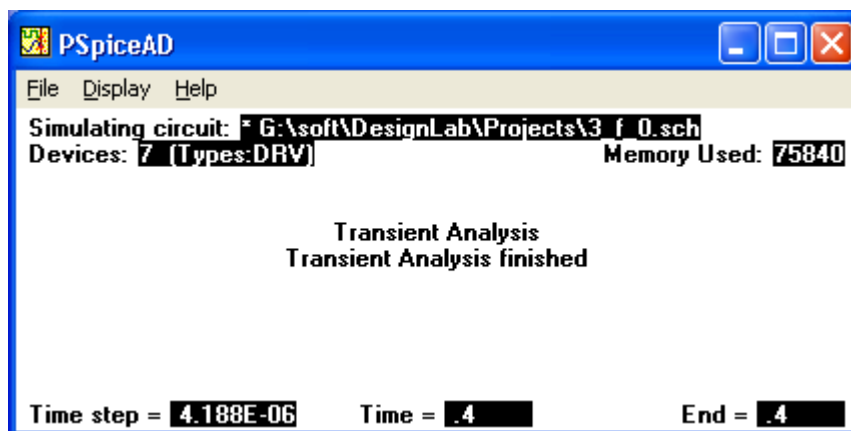


Рисунок 4.3 – Вікно програми PSpice

Для виведення результатів моделювання використовують програму Probe. Вона виводить на екран графіки результатів моделювання, проводить їх математичну обробку і виводить на екран у формі таблиці їх найважливіші характеристики, наносить на графіки пояснюючі написи і дозволяє отримувати жорсткі копії результатів моделювання в графічній формі.

Математичні перетворення над графіками полягають у виконанні арифметичних операцій, обчисленні різних функцій, взятті інтегралів, розрахунку спектрів, гістограм, вимірів, параметрів форми графіків, побудові залежностей будь-якої характеристики графіку від будь-якого змінного параметра схеми.

Програму Probe викликають автономно або під управлінням Schematics вибором команди Analysis \ Probe або натисканням клавіші *[F12]* (попередньо рекомендують виконану програму моделювання PSpice поточної схеми закрити). Під час виклику Probe з програми Schematics можна організувати екран з кількома вікнами для зображення схеми і графіків різних характеристик.

У вікні програми Probe зображені графіки перехідних процесів і частотних характеристик. Змінні, графіки яких має бути виведено на екран, позначають відміткою їх спеціальними маркерами на вікні схеми. Можливість перемикання вікон зі схемою і графіками істотно полегшує осмислення результатів моделювання (рис.4.4).

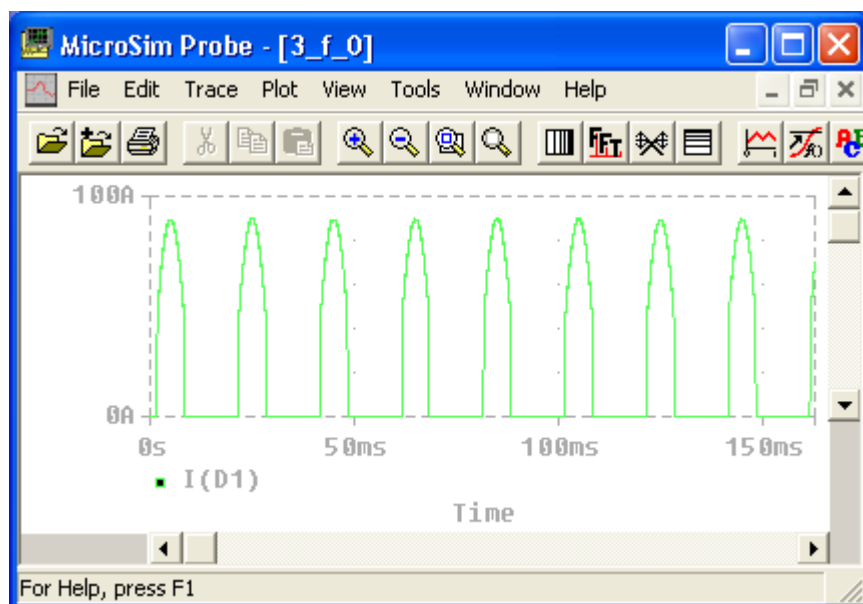



Рисунок 4.4 – Вікно програми Probe


На вікні схеми можна зробити позначку маркером будь-якого ланцюга або виводу компонента, і тоді у вікні програми Probe негайно буде побудовано

відповідний графік. У рядку меню є електронний курсор , який дозволяє під час його руху виводити значення всіх змінних. У програмі Schematics є кілька маркерів (напруги, струму і ін.), які розташовані у меню Markers. Маркер Voltage \ Level вимірює напругу в будь-якій точці схеми відповідно до заземленого провідника, маркер Voltage Differential вимірює напругу між двома точками, маркер Current into pin вимірює струм в ділянці схеми (цей маркер встановлюють на схемі тільки в контакт з елементом схеми, без дрітів).

### **Порядок виконання роботи.**

Робота в *Design Lab* на прикладі моделювання простої схеми

Роботу з системою *Design Lab* зазвичай починають зі створення принципової схеми за допомогою редактора схем *Schematics*. Він дозволяє створювати креслення принципів схем і здійснювати запуск інших програм (*PSpice*, *Probe* і ін.). Модельована схема може мати у своєму складі наступні типові компоненти: резистори, конденсатори, індуктивності, трансформатори (в тому числі з магнітними сердечниками), діоди (включаючи стабілітрони і варикапи), біполярні, арсенід-галієві, польові, МОП – і біполярні статично індуковані транзистори, ключі, керовані струмом і напругою, лінії передачі з втратами, аналого-цифрові та цифро-аналогові перетворювачі, цифрові елементи (вентилі, тригери, пристрої контролю, що запам'ятовують пристрої і програмовані логічні матриці).

Створення схеми починаємо з розміщення компонентів. Вибір компонента можна здійснити з алфавітного списку в окремій бібліотеці або шляхом пошуку за ключовими словами за всіма компонентами. Найбільш зручним є спосіб вибору компонентів з відповідних бібліотек. Для цього слід натиснути кнопку вибору компонентів  або виконати команду Draw / Get New Part. У вікні, що розкрилося Part Browser Basic натиснути кнопку Libraries (бібліотеки), у полі Library обрати бібліотеку Analog.slb (аналогові компоненти), а в полі Part – потрібний компонент (рис. 4.5).

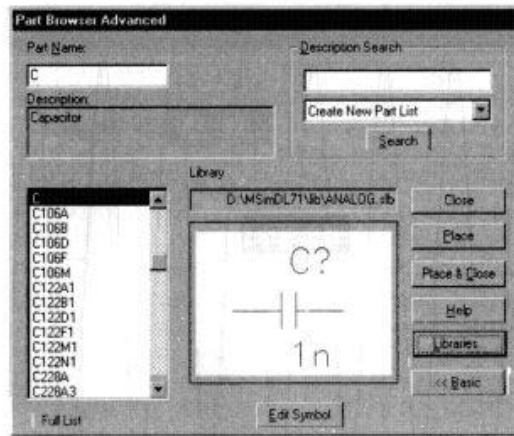


Рисунок 4.5

Для створення простої схеми, вибираємо елементи з бібліотеки і поєднуємо їх у необхідній послідовності (рис.4.6).

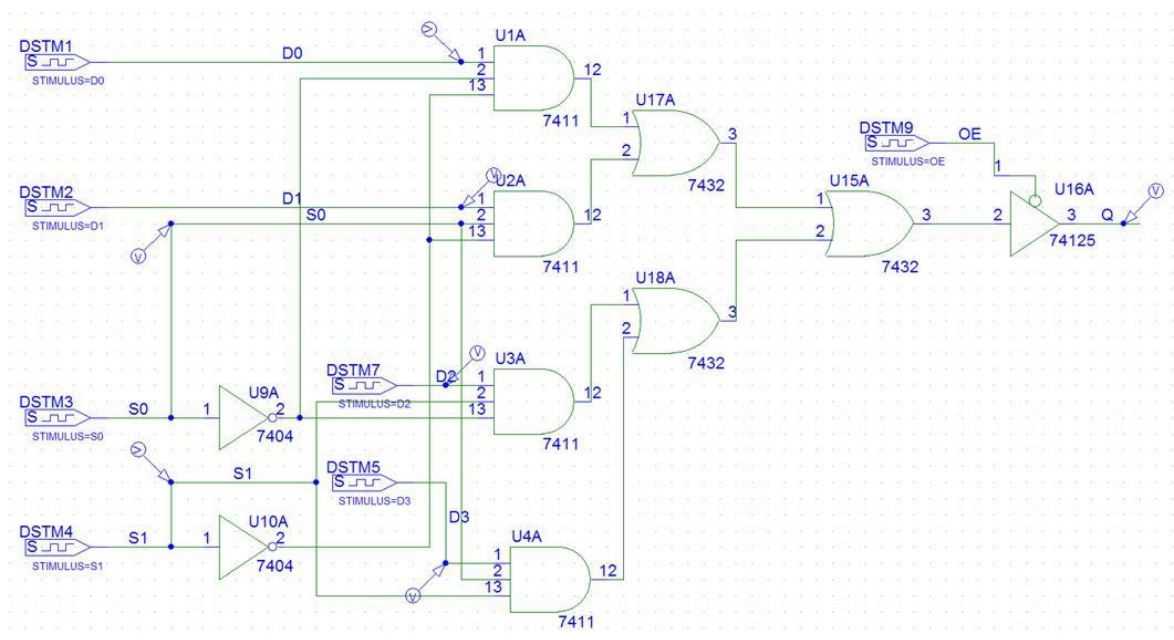


Рисунок 4.6 – Цифрова схема пристрою

Для моделювання процесу роботи, подаємо на схему входні впливи за допомогою стимулів (DigStim). Входні впливи та інтервал моделювання схеми вибираємо такі (рис. 4.7), щоб можна було оцінити якість роботи схеми, адекватність моделі. Час моделювання 2.2 мкс і входні впливи задані програмою Stimulus Editor.

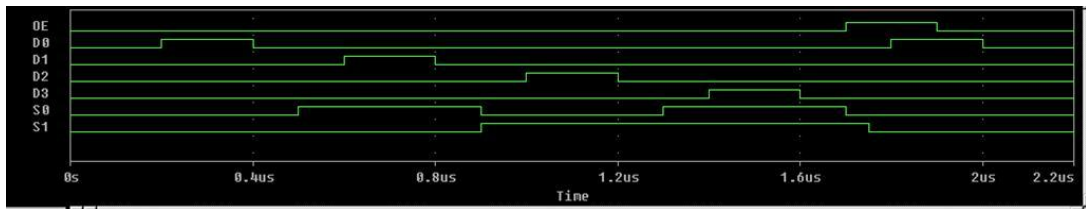


Рисунок 4. 7 – Завдання вхідних впливів на модель

Розставляємо маркери для зняття параметрів роботи моделі, також за допомогою постпроцесора моделювання Probe об'єднуємо безліч зв'язкових сигналів D3, D2, D1, D0 і S1, S0 в шини DBUS і Address для спрощення сприйняття результатів моделювання (рис. 4.8).

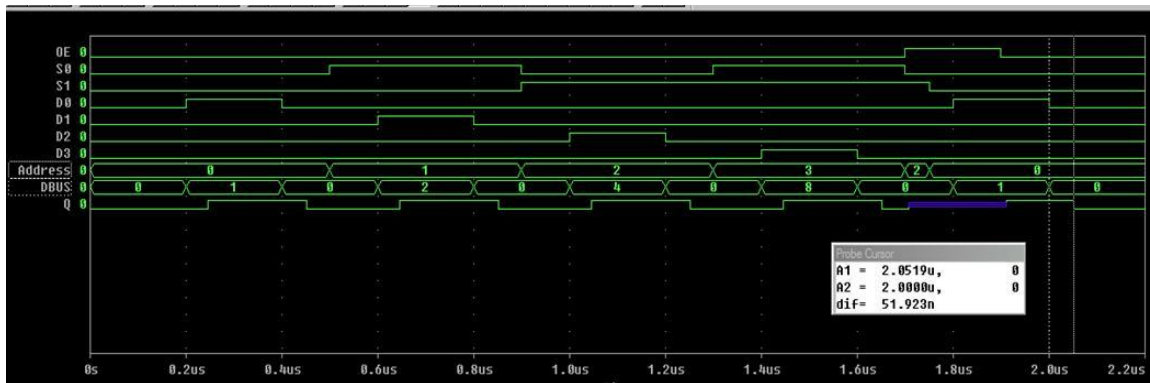


Рисунок 4.8 – Тимчасова діаграма роботи пристрою

За діаграмою можна побачити, що модель працює згідно з її ідеєю і затримка реакції вихідного сигналу на перемикання шини даних дорівнює 51.93нс, аналогічно можна виміряти інші затримки. Також з моделювання видно, що затримка перемикання вихідного сигналу з 1 в 0 не дорівнює затримці перемикання з 0 в 1

### *Проектування на корпусних мікросхемах*

Перший етап проектування (рис.4.9) – створення вхідного опису проекту (Entry Design).

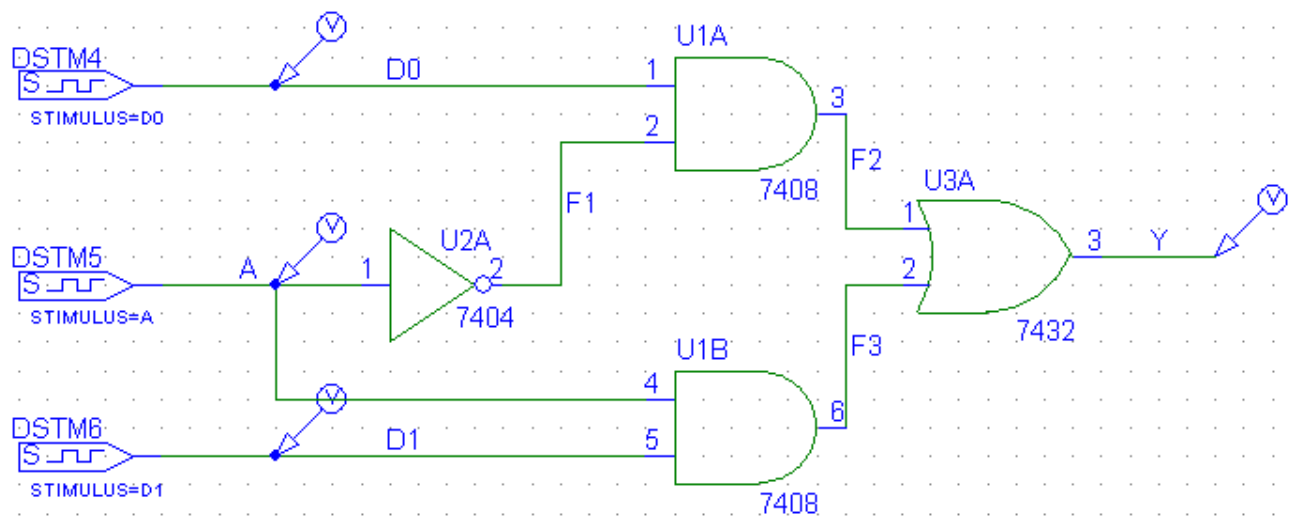


Рисунок 4.8

На другому етапі виконують логічне моделювання (функціональна верифікація) з метою перевірки правильності роботи (рис.4.9).

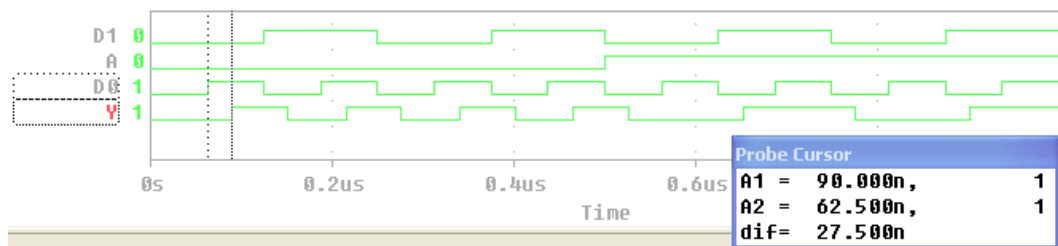


Рисунок 4.9

На тимчасових діаграмах можемо побачити затримку поширення сигналу. В даному випадку, виміряна затримка від входу D0 до виходу Y. Вона становить 27.5 ns.

На третьому етапі верифіковану схему треба підготувати для створення друкованої плати (рис.4.10).

Зокрема, зі схеми видаляють генератори вхідних сигналів (стимули), і замість них встановлюють роз'єми.

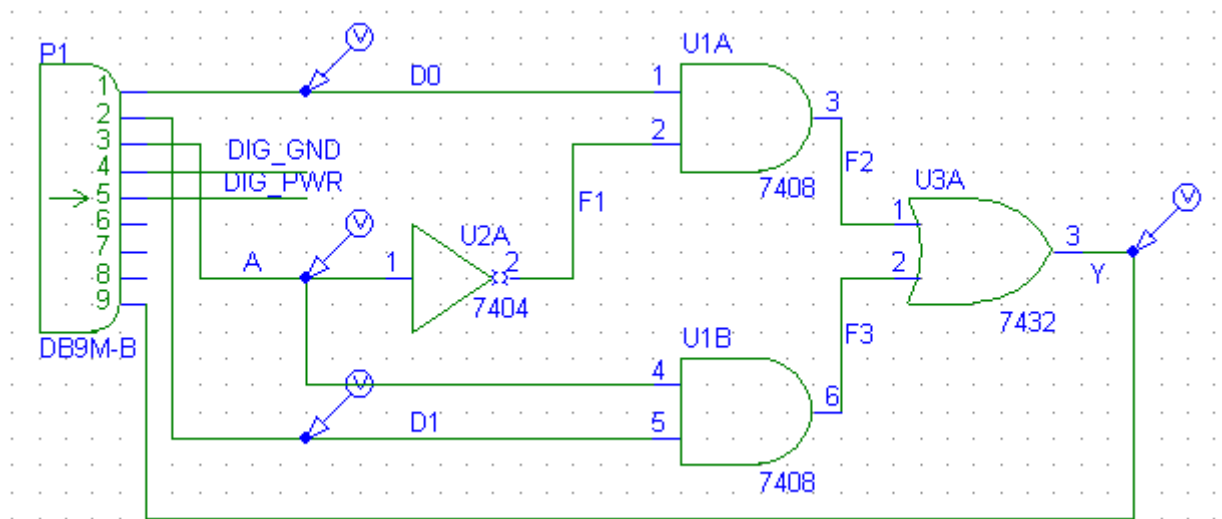


Рисунок 4.10

Трасування друкованої плати виконують на четвертому етапі в автоматичному режимі (рис.4.11).

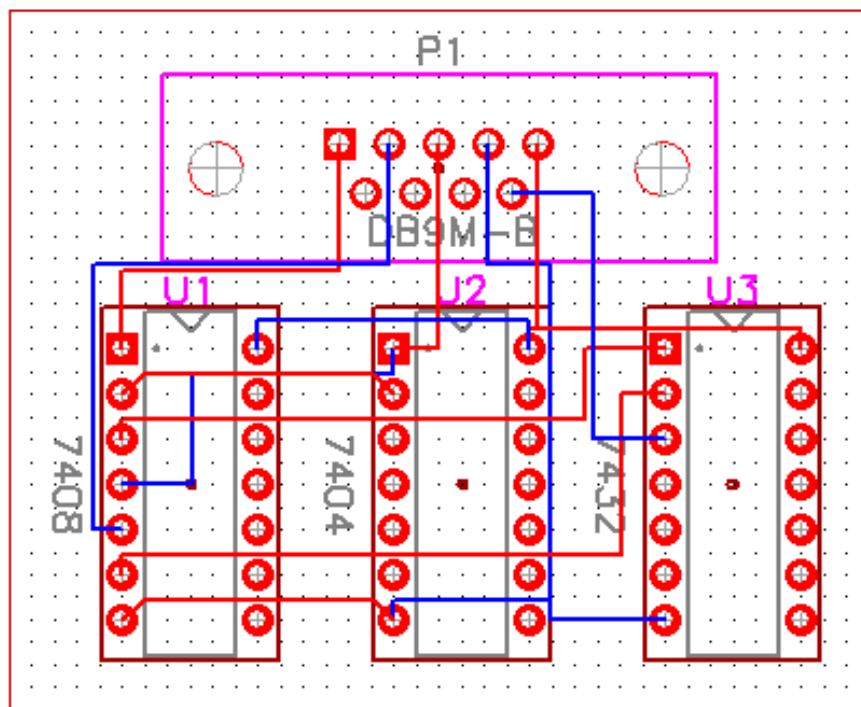


Рисунок 4.11

За допомогою даного пакету можна досліджувати роботу схем на різних швидкостях, що є дуже зручно. Програма видає всі порушення у вигляді повідомлень, таким чином можна відстежувати повільні елементи в схемі і покращувати її якість.

В цілому продукт є дуже корисним для моделювання, він забезпечує наскрізне проектування і зручний на кожному рівні проектування.



**КОНТРОЛЬНІ ПИТАННЯ ЗА ТЕМОЮ:**  
**«Інформаційні системи технічного моделювання та конструювання»**

1. Що таке «технічне моделювання»?
2. Класифікація технічних систем в моделюванні.
3. Поясніть відмінність між геометричними та фізичними моделями досліджуваних систем.
4. Способи і методи технічного моделювання.
5. Класифікація систем технічного моделювання.
6. Сфери використання систем технічного моделювання.
7. Як створити нову модель в інформаційних технічних системах моделювання?
8. Як з'єднати два блоки, що мають відповідно вільний вихід і вільний вхід?
9. Як зробити, щоб один і той самий сигнал надходив на кілька блоків?
10. Як передати результати моделювання в робочу область *MATLAB*? У якому вигляді вони передаються?
11. Що визначається в процесі аналізу системи?
12. Що визначається в процесі синтезу системи?
13. Чим оцінюється ефективність системи?
14. Що розуміється під оптимальною системою?
15. Властивості, притаманні складній системі, і їх коротка характеристика.
16. Як з'єднувати і роз'єднувати блоки?
17. Чим відрізняється реалізація функціональної організації системи від структурної?
18. Для чого потрібні списки з'єднань?
19. Способи підключення бібліотек компонентів.
20. Методи автоматичного трасування компонентів.
21. Відмінності, переваги та недоліки різних систем технічного моделювання.
22. У яких випадках технічне моделювання виправдане і необхідно?
23. Який метод дослідження систем є найбільш точним?
24. У чому полягають принципи функціонування програми *VisSim*?
25. Призначення і особливості програмного середовища *Labview*.

## СПИСОК ЛІТЕРАТУРИ

1. Информационные системы конструирования и моделирования объектов. Навчальн. посібник / . Адашевская И.Ю. / . Харків: «НТМТ», 2016.– 200 с.
2. Уваров А.С. Проектирование и конструирование электронных устройств. – М.: Юнити-Дана, – 2004.
3. Основы компьютерного проектирования и моделирования РЭС. Лабораторный практикум / П.Б.Абрамов, Л. Б. Афанасьевский, А.Н.Горин, А.Г. Фадин; под ред. проф. А.Г.Фадина. – Воронеж : ВИРЭ, 2002. – 268 с.
4. Борисов Ю.П. Математическое моделирование радиотехнических систем и устройств / Ю.П. Борисов, В.В. Цветнов– М.: Радио и связь, 1985. – 176 с.
5. LabVIEW: практикум по основам измерительных технологий. / В.К. Батоврин, А.С.Бессонов, В.В. Мошкин и др. Мн. : Выш. шк., – 2010.
6. Сениченков Ю.Б. Численное моделирование гибридных систем / Ю.Б. Сениченков, 2004.
7. Моделирование систем в программе VisSim: Справочная система / Н.В. Клиначёв, 2003.
8. Алексеева О.В. Автоматизация проектирования радиоэлектронных средств : учеб. пособ. для студентов вузов. / О.В. Алексеева: под ред. О.В. Алексеева. – М.: Высшая школа, 2000. – 479 с.
9. Фрике К. Вводный курс цифровой электроники / К. Фрике – М: Техносфера, 2003. – 432 с.
- 10.Кулинич Ю.М. Современная силовая электроника : учеб. пособие / Ю.М.Кулинич. – Хабаровск : Изд-во ДВГУПС, 2006. – 95 с.

ДЛЯ ПРИМІТОК

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

**Методи конструювання об'єктів в комп'ютерних системах  
(Частина II)**

для студентів спеціальності 122 «Комп'ютерні науки»

Укладач: АДАШЕВСЬКА Ірина Юріївна

За авторською редакцією

План 2018 р., поз. 70 .

Підписано до друку 24.05.18 .      Формат 60×84 1/16.

Папір друк. № 2.

Друк-ризографія. Гарнітура Times New Roman. Ум. друк. арк.

Наклад 50 прим. Зам. № 12. Ціна договірна.

---

Видавничий центр НТУ «ХПІ». 61002, Харків, вул. Кирпичова, 2.

Свідоцтво про реєстрацію ДК № 3657 від 24.12.2017 р.

---

Друкарня НТУ «ХПІ», 61002, Харків, вул. Кирпичова, 2.